HTTP

Módulo 1

Bruno Mendoza Guedes
http://www.brunomendoza.es
hola@brunomendoza.es
@_brunomendoza

Introducción

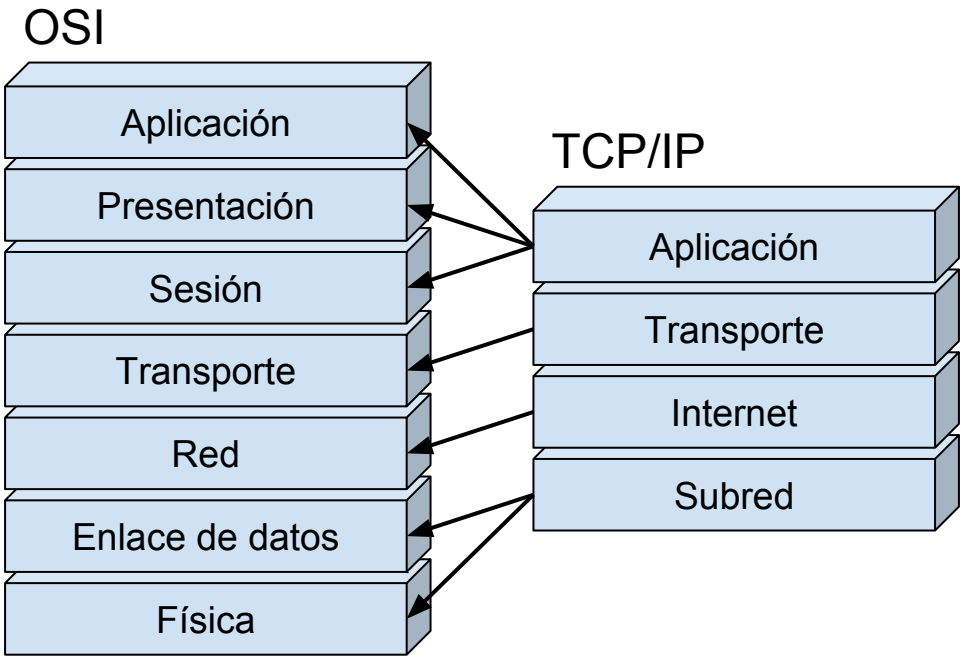
Introducción

- Acrónimo de **Hypertext Transfer Protocol**
- Permite traer recursos.
- Base de cualquier intercambio de datos en la Web. Principalmente entre navegadores Web y servidores Web.
- Peticiones iniciadas por el emisor (cliente-servidor).
- Un documento completo se reconstruye a partir de otros sub-documentos capturados (hypermedia).

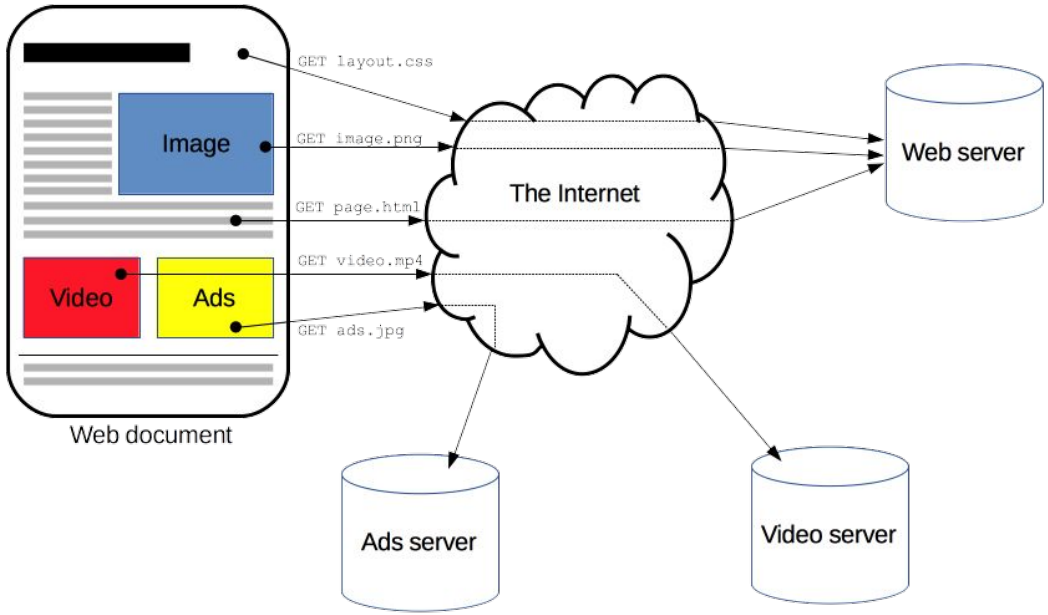
Introducción

- Clientes y servidores se comunican intercambiando mensajes individuales.
- Los mensajes enviados por el cliente se llaman **request** (petición).
- Los mensajes enviados por el servidor se llaman **response** (respuesta).
- Diseñado en los 90.
- Protocolo de capa de aplicación enviado sobre una conexión TCP o TCP encriptado con TLS (Transport Layer Security). Comunicación orientada a conexión.

TCP/IP contra OSI



Ejemplo

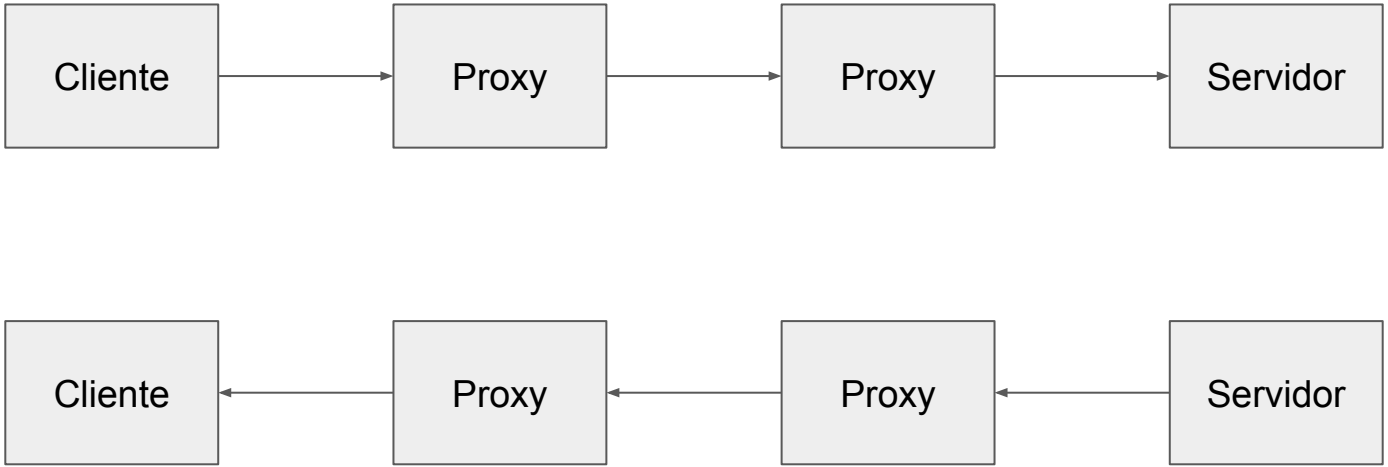


Componentes

Componentes

- HTTP es un protocolo cliente-servidor.
- El cliente se denomina **user-agent** o proxy.
- Cada **petición** (request) se envía a un servidor.
- El servidor procurará una **respuesta** (response).
- Entre la petición y la respuesta hay múltiples entidades llamadas **proxies**. Por ejemplo, puertas de enlace o cache.
- HTTP se encuentra en lo más alto de la capa de aplicación.

Componentes



Cliente: el agente de usuario

- El agente de usuario es una herramienta que actúa en nombre del usuario.
- El navegador es siempre la entidad que comienza la petición.
- Para presentar una página Web, el navegador envía petición para traer el documento HTML desde el servidor.
- El navegador traduce el archivo, ejecutando el resto de peticiones.
- El navegador mezcla estos recursos para para presentar al usuario el resto el documento completo.
- Una página Web es un documento hipertextual (hypertext document).

El servidor Web

- Sirve los documentos requeridos por el cliente.
- Puede ser una colección de servidores.
  - Para compartir la carga (load balancing).
  - Una pieza compleja de software interrogando a otras computadoras (cache, servidor de bases de datos o servidores e-commerce).

Proxies

- Entre el navegador Web y el servidor hay múltiples computadoras y máquinas que reenvían (relay) los mensajes.
- La mayoría operan en la capa de transporte, red física, siendo transparente para la capa HTTP.
- Las que operan sobre la capa de aplicación se llaman **proxies**.
  - Caching. Cache pública o privada, como la cache del navegador.
  - Filtrado. Como un escáner de antivirus o control parental.
  - Balanceo de carga. Control de acceso a los recursos.
  - Logging. Permitir almacenar información histórica.

Aspectos Básicos de HTTP

HTTP es simple

- HTPP está diseñado para que sea legible para los humanos.

HTTP es extensible

- Este protocolo es fácil de extender y de experimentar con él.
- Se pueden introducir nuevas funcionalidades con un simple arreglo entre cliente y servidor sobre el significado de las nuevas cabeceras.

HTTP es *stateless*, pero no *sessionless*

- No hay enlace entre dos peticiones sucesivas llevadas sobre la misma conexión.
- Mientras el núcleo de HTTP es *stateless*, las *cookies* HTTP permiten el uso de sesiones *stateless*.
- Através de la extensibilidad de las cabeceras, las *cookies* HTTP se añaden al flujo de trabajo, permitiendo la creación de sesiones sobre cada petición HTTP para compartir el mismo contexto o el mismo estado.

HTTP y las conexiones

- Una conexión es controlada en la capa de transporte y, por lo tanto, fuera del alcance de HTTP.
- Es necesario un protocolo *de confianza* y no obligatoriamente orientado a conexión.
- TCP es *de confianza* y UDP no lo es.
- HTTP/1.0 abre una conexión por cada intercambio entre petición y respuesta.
- HTTP/1.1 introdujo el entubado (pipelining) y las conexiones persistentes.
    - La conexión TCP subyacente puede ser controlada parcialmente a través de la cabecera *Connection*.
- HTTP/2 multiplexa los mensajes en una conexión simple, ayudando a mantener la conexión *caliente* y mejorar la eficiencia.

Control HTTP

Control HTTP

- La extensibilidad de HTTP permite más control y funcionalidades sobre la Web.
- Método de cache y autenticación fueron funciones manejadas al comienzo de la historia de HTTP. Sin embargo, el relajamiento *restricciones del origen* se introdujo en años posteriores al 2010.

Características

- Cache. El protocolo HTTP puede controlar cómo se *cachean* los documentos.
- Relajamiento de las restricciones de origen. Las cabeceras web permiten a un documentos convertirse en un *patchwork* de información traída de diferentes dominios.
- Autenticación. HTTP puede proporcionar autenticación básica.
- Proxy y *tunneling*. Los servidores y/o clientes están a veces localizados dentro de intranets y ocultan sus direcciones IP a otros. La peticiones HTTP atraviesan entonces *proxies* para cruzar esta barrera de red.
- Sesiones. El uso de cookies permite enlazar peticiones con el estado del servidor.

Flujo HTTP

Flujo HTTP

Pasos entre la comunicaciones cliente-servidor:

1. Abrir la conexión TCP.
2. Enviar un mensaje HTTP.
3. Leer la respuesta enviada por servidor.
4. Cerrar la conexión.

Abrir la conexión TCP

- La conexión TCP se usará para enviar una petición, o varias, y recibir una respuesta.
- El cliente puede abrir una nueva conexión, reutilizar una conexión existente o abrir varias conexiones TCP al servidor.

Enviar un mensaje HTTP

- Las cabeceras HTTP (antes de HTTP/2) son legibles por los humanos. Con HTTP/2 estos mensajes simples son encapsulados en *frames* (marcos).

```
GET / HTTP/1.1
Host: developer.mozilla.org
Accept-Language: es
```

Leer la respuesta

```
HTTP/1.1 200 OK
DATE: Mon, 01 Dec 2017 14:20:05 GMT
Server: Apache
Last-Modified: Tue, 01 Nov 2010 20:20:00 GMT
ETag: “51142bc1-7449-479b075b2891b”
Accept-Ranges: bytes
Content-Length: 29896
Content-Type: text/html

<!DOCTYPE html ...
```

Cerrar la conexión

O reutilizarla para realizar más peticiones.

Nota

Si el *pipelining* está activado, algunas peticiones pueden ser enviadas sin esperar que la primera respuesta sea completamente recibida.

Esta tecnología ha sido sustituida por peticiones multiplexadas más robustas dentro de un *frame*.

Mensajes HTTP

Mensajes HTTP

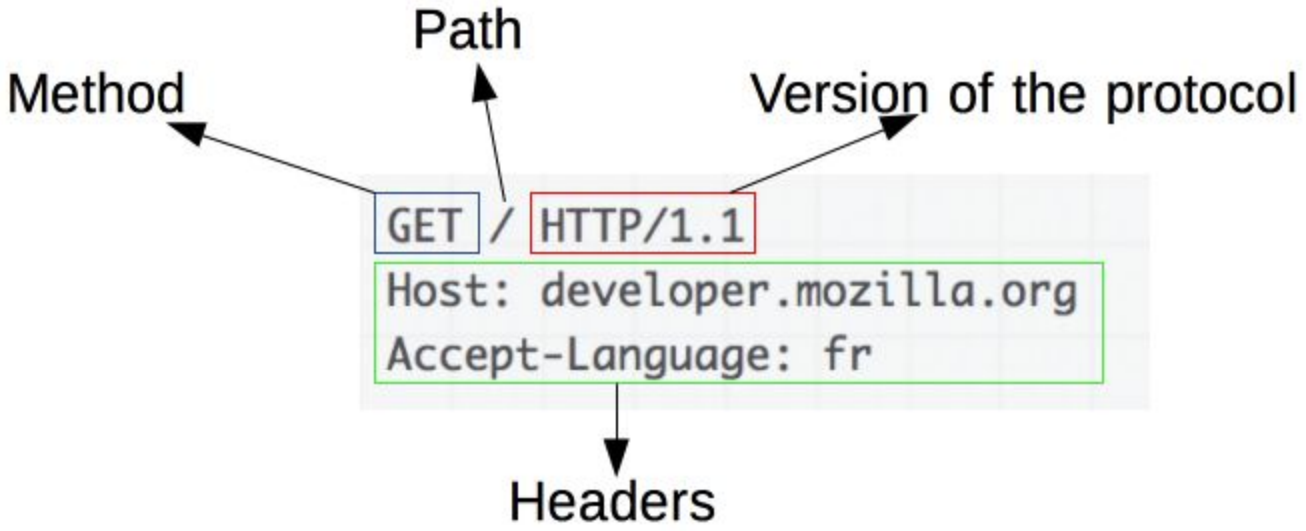
- Los mensajes HTTP/1.1 y anteriores son legibles por los humanos.
- Los mensajes HTTP/2 esos mensajes son embebidos en una nueva estructura binaria, un *frame*.
- El *frame* permite optimizaciones como la compresión de las cabeceras y el multiplexado.
- Existen dos tipos de mensajes HTTP: peticiones y respuestas.

Peticiones

Una petición consiste en los siguiente elementos:

- Un método
- La ruta del recurso. La URL del recurso separada de los elementos obvios dentro de ese contexto.
- La versión del protocolo HTTP.
- Cabeceras opcionales.
- Un cuerpo para métodos como POST que contienen el recurso enviado.

Ejemplo de petición

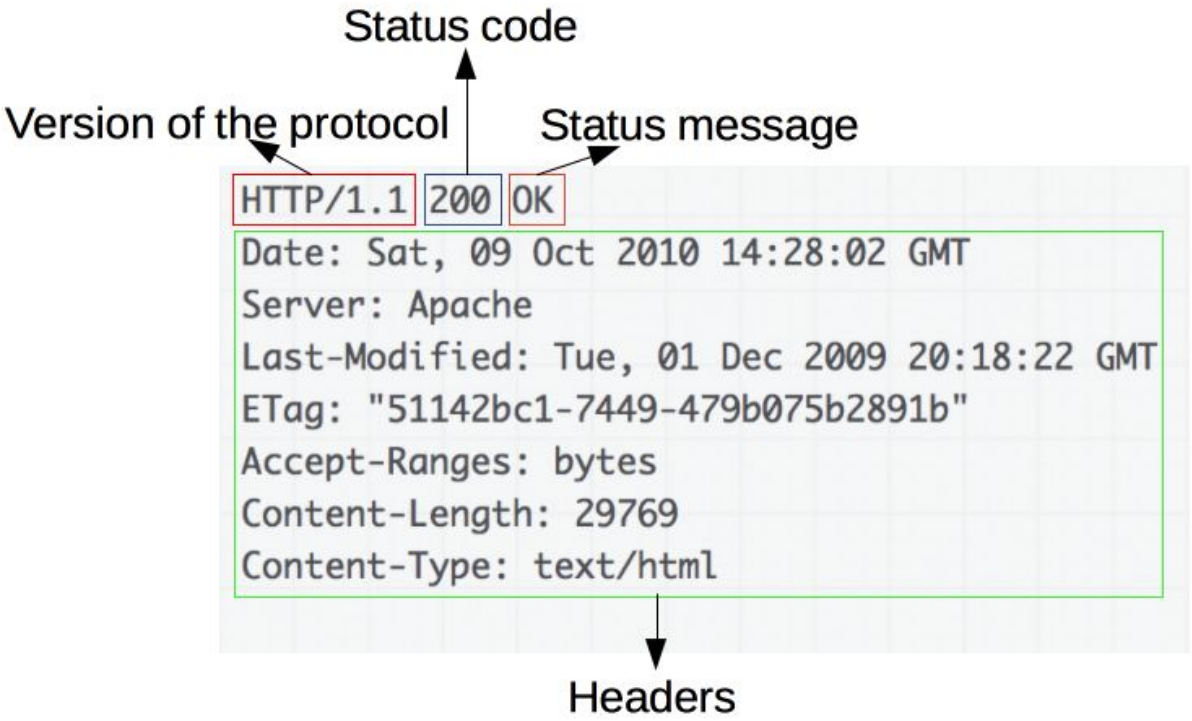


Respuestas

Las respuestas consisten en los siguientes elementos:

- La versión del protocolo HTTP.
- El código de estado.
- Un mensaje de estado.
- Cabeceras HTTP.
- Y opcionalmente un cuerpo (body) que contiene el recurso solicitado.

Ejemplo de respuesta



Métodos HTTP

Métodos

HTTP define un conjunto métodos para indicar la acción deseada que será ejecutada para un recurso dado:

- **GET** sirve para obtener un recurso.
- HEAD igual que GET pero sin cuerpo.
- **POST** sirve para enviar una entidad para el recurso determinado.
- **PUT** sustituye todas las representaciones actuales de un recurso objetivo.
- **DELETE** eliminar el recurso especificado.
- CONNECT establece un túnel al servidor por el recurso objetivo.
- OPTIONS es utilizado para describir las opciones de comunicación.
- TRACE realiza un prueba de *loop-back* de un mensaje a lo largo de la ruta del recurso objetivo.
- PATCH es utilizado para realizar modificaciones parciales a un recurso.

Conclusión

Conclusión

- HTTP es un protocolo extensible fácil de usar.
- La estructura cliente-servidor, combinada con la capacidad de añadir cabeceras, permiten a HTTP avanzar con la capacidades extendidas de la Web.
- Aunque HTTP/2 añada alguna complejidad la estructura básica de los mensajes se ha mantenido desde HTTP/1.0.
- El flujo de sesión permanece simple permitiendo ser investigado y depurado con un monitor de mensajes HTTP simple.